Systematic feedback (recursive) encoders

- \( G'(D) = [1, (1 + D^2) / (1 + D + D^2), (1 + D) / (1 + D + D^2)] \)
- Infinite impulse response (not polynomial)
  - Easier to represent in transform domain
- Sometimes referred to as systematic recursive convolutional code (SRCC). But note that the feedback is an encoder property
Example

\[
G = \begin{pmatrix}
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 \\
& & & & & \vdots
\end{pmatrix}
\]

\[
G(D) = \begin{pmatrix}
1+D & D & 1+D \\
D & 1 & 1 \\
1+D/(1+D) & D/(1+D) & 1+D/(1+D) \\
D & 1 & 1 \\
\end{pmatrix}
\]

\[
\rightarrow \begin{pmatrix}
1 & D/(1+D) & 1 \\
D+D & 1+D^2/(1+D) & 1+D \\
\end{pmatrix}
\rightarrow \begin{pmatrix}
1 & D/(1+D) & 1 \\
0 & (1+D+D^2)/(1+D)(1+D+D^2) & (1+D)/(1+D+D^2) \\
\end{pmatrix}
\rightarrow \begin{pmatrix}
1 & 0 \frac{1}{1+D+D^2} \\
0 & 1 \frac{1+D^2}{1+D+D^2}
\end{pmatrix}
\]
Example (cont.)

Controller canonical form

\[
G'(D) = \begin{pmatrix}
1 & 0 & \frac{1}{1+D+D^2} \\
0 & 1 & \frac{1+D^2}{1+D+D^2} \\
\end{pmatrix}
\]

\[
H'(D) = \begin{pmatrix}
\frac{1}{1+D+D^2} & 1+D^2 & 1 \\
1+D+D^2 & 1+D+D^2 & 1 \\
\end{pmatrix}
\]

Observer canonical form

\[
G'(D) = \begin{pmatrix}
1 & 0 & \frac{1}{D^2+D+1} \\
0 & 1 & \frac{D^2+1}{D^2+D+1} \\
\end{pmatrix}
\]

\[
H'(D) = \begin{pmatrix}
\frac{1}{D^2+D+1} & \frac{D^2+1}{D^2+D+1} & 1 \\
\end{pmatrix}
\]
Some notes on minimality

a) For rate $1/n$ codes, there is a minimal encoder is in controller canonical form provided common factors are removed from $G(D)$ (or $G'(D)$)
   • Why:
     • $k = 1$, only one row of polynomials
     • If common factors are not removed:
       • $G(D) = [1 + D^2, 1 + D^3]$ ($\nu=3$), but after dividing by $1+D$
       • $[1 + D, 1 + D + D^2]$ ($\nu=2$)

b) For rate $(n-1)/n$ codes, there is a minimal encoder in observer canonical form corresponding to a systematic recursive encoder $G'(D)$, provided
   • all common factors are removed from $H(D)$, corresponding to the systematic rational function parity check matrix $H'(D)$ derived from $G'(D)$, and
   • all functions in $G'(D)$ are realizable (delay of denominator exceeds delay of numerator)
   • Why:
     • $k = n - 1$, only one row of polynomials in $H(D)$

c) For rate $k/n$: More difficult
**Example: Nonsystematic feedback encoder**

\[
G(D) = \begin{bmatrix}
\frac{1}{1+D+D^2} & \frac{D}{1+D^3} & \frac{1}{1+D^3} \\
\frac{D^2}{1+D^3} & \frac{1}{1+D^3} & \frac{1}{1+D}
\end{bmatrix}
\]

We need a common denominator in each row for the feedback of the shift registers:

\[
G'(D) = \begin{bmatrix}
\frac{(1+D)}{(1+D^3)} & \frac{D}{1+D^3} & \frac{1}{1+D^3} \\
\frac{D^2}{1+D^3} & \frac{1}{1+D^3} & \frac{1+D+D^2}{1+D^3}
\end{bmatrix}
\]

Controller canonical form

Observer canonical form
Summary

a) Any rate $1/n$ code can be represented by a minimal feedforward encoder $G(D)$, or its rational systematic counterpart

b) Any rate $(n-1)/n$ code can be represented by a minimal systematic feedback matrix $H(D)$, or its polynomial counterpart
Feedforward encoder inverse

a) What if we want an encoder inverse; i.e. a way to find an estimate of the information by inverting the received word without decoding?

b) Systematic encoders: Easy. Nonsystematic encoders?

c) Recall: \( V(D) = U(D) G(D) \)

d) Then, we need an inverse matrix \( G^{-1}(D) \) such that \( G(D) G^{-1}(D) = I D' \)

e) Then, \( V(D) G^{-1}(D) = U(D) G(D) G^{-1}(D) = U(D) I D' = U(D) D' \)

f) When does a feedforward inverse exist?

g) Rate 1/n code: A feedforward inverse exists iff
   - \( \text{GCD} \{ g^{(0)}(D), g^{(1)}(D), ..., g^{(n-1)}(D) \} = D' \)

h) Rate k/n code: A feedforward inverse exists iff
   - \( \text{GCD} \{ \Delta_i(D) \} = D' \)
   - \( \text{where } \{ \Delta_i(D) \} \text{ is the set of all determinants of } k \times k \text{ submatrices} \)
Example

a) \( G(D) = [1 + D^2 + D^3, 1 + D + D^2 + D^3] \)

b) GCD \( \{1 + D^2 + D^3, 1 + D + D^2 + D^3\} = 1 \)

\[ G^{-1}(D) = \begin{pmatrix} 1 + D + D^2 \\ D + D^2 \end{pmatrix} \]
Example

\[
G(D) = \begin{pmatrix}
1 + D & D & 1 + D \\
D & 1 & 1 \\
1 & 1 + D & D
\end{pmatrix}
\]

\[
\text{GCD}\left\{1 + D + D^2, 1 + D + D + D^2, D + 1 + D\right\} = 1
\]

\[
G^{-1}(D) = \begin{pmatrix}
0 & 0 \\
1 & 1 + D \\
1 & D
\end{pmatrix}
\]
Example 11.10: Catastrophic encoder

a) \( G(D) = [1 + D, 1 + D^2] \)
b) \( \text{GCD} \{1 + D, 1 + D^2\} = 1 + D \)
c) There is no feedforward inverse
d) Example: \( U(D) = 1 / (1 + D) = 1 + D + D^2 + D^3 + \ldots \)
   • will correspond to the codeword \( V(D) = U(D) \cdot G(D) = 1 / (1 + D) \cdot [1 + D, 1 + D^2] = (1, 1 + D) \)
   • Infinite input weight implies finite output weight
   • A catastrophic encoder
e) Must not be used
f) Minimal encoders and systematic encoders are non-catastrophic
Structural properties

a) From now on we assume a minimal encoder

b) The state diagram:
   • **State:** Describes the content of the encoder’s memory elements
     • Implies a natural state labeling
   • **State transitions:** Describe what happens when a $k$-bit input block arrives when the encoder is in a given state
     • **Direction:** What will be the new state?
     • **Label:** Which $n$ output bits are produced?
Example
Example
Example: Catastrophic encoder

a) \( G(D) = [1 + D, 1 + D^2] \)

b) \( \text{GCD} \{1 + D, 1 + D^2\} = 1 + D \)
Weight enumerator function

a) A codeword weight enumerating function (WEF) is an expression that completely describes the weights of all codewords.

b) The WEF is a code (not encoder specific) property.

c) The WEF can be derived by inspection of the state diagram with some enhancements as follows:
   - Split state $S_0$ into an initial and a terminal state.
   - The zero-weight branch around state $S_0$ is deleted.
   - Label each branch by $X^d$, where $d$ is the Hamming weight of the $n$-bit output block corresponding to the branch.

d) The path gain of a particular path is the product of the branch gains over the path. Thus, the Hamming weight of the path is the power of $X$ in the path gain.
Modified state diagrams for WEF determination
Modified state diagrams for WEF determination
Mason’s gain formula

- Want to compute the transfer function $A(X) = \sum_d A_d X^d$
- Definitions:
  - Forward path: Path going from initial to final state, touching each state at most once. There is a finite number of such paths
    - $F_i =$ gain of the $i$th forward path
  - Cycle: Path starting at one state and coming back to the same state, touching any other state at most once. There is a finite number of such cycles. $C_i =$ gain of the $i$th cycle
    - $\Delta = 1 - \sum_i C_i + \sum_{i,j} C_i C_j - \sum_{i,j,l} C_i C_j C_l + \ldots$
      (sum over non-intersecting cycles)
    - $\Delta_m = 1 - \sum_i C_i^{(m)} + \sum_{i,j} C_i^{(m)} C_j^{(m)} - \sum_{i,j,l} C_i^{(m)} C_j^{(m)} C_l^{(m)} + \ldots$
      (sum over non-intersecting cycles not touching the $m$th forward path)
    - $A(X) = \sum_i F_i \Delta_i / \Delta$
Example 11.12

Forward Path 1: $S_0S_1S_3S_7S_6S_5S_2S_4S_0 \quad (F_1 = X^{12})$
Forward Path 2: $S_0S_1S_3S_7S_6S_4S_0 \quad (F_2 = X^7)$
Forward Path 3: $S_0S_1S_3S_6S_5S_2S_4S_0 \quad (F_3 = X^{11})$
Forward Path 4: $S_0S_1S_3S_6S_4S_0 \quad (F_4 = X^6)$
Forward Path 5: $S_0S_1S_2S_5S_3S_7S_6S_4S_0 \quad (F_5 = X^3)$
Forward Path 6: $S_0S_1S_2S_5S_3S_6S_4S_0 \quad (F_6 = X^7)$
Forward Path 7: $S_0S_1S_2S_4S_0 \quad (F_7 = X^7)$

$\Delta = 1 - (X^8 + X^3 + X^7 + X^2 + X^4 + X^3 + X + X^5 + X^4 + X)$
$\quad + (X^4 + X^8 + X^3 + X^4 + X^8 + X^7 + X^4 + X^2 + X^5) - (X^4 + X^8)$
$\quad = 1 - 2X - X^3. \quad (11.100)$

Cycle 1: $S_1S_3S_7S_6S_5S_2S_4S_1 \quad (C_1 = X^8)$
Cycle 2: $S_1S_3S_7S_6S_4S_1 \quad (C_2 = X^3)$
Cycle 3: $S_1S_3S_6S_5S_2S_4S_1 \quad (C_3 = X^7)$
Cycle 4: $S_1S_3S_6S_4S_1 \quad (C_4 = X^2)$
Cycle 5: $S_1S_2S_5S_3S_7S_5S_4S_1 \quad (C_5 = X^4)$
Cycle 6: $S_1S_2S_5S_3S_6S_4S_1 \quad (C_6 = X^3)$
Cycle 7: $S_1S_2S_4S_1 \quad (C_7 = X^3)$
Cycle 8: $S_2S_5S_2 \quad (C_8 = X)$
Cycle 9: $S_3S_7S_6S_5S_3 \quad (C_9 = X^5)$
Cycle 10: $S_3S_6S_5S_3 \quad (C_{10} = X^4)$
Cycle 11: $S_7S_7 \quad (C_{11} = X)$

Cycle Pair 1: (Cycle 2, Cycle 8) \quad (C_2C_8 = X^4)
Cycle Pair 2: (Cycle 3, Cycle 11) \quad (C_3C_{11} = X^8)
Cycle Pair 3: (Cycle 4, Cycle 8) \quad (C_4C_8 = X^3)
Cycle Pair 4: (Cycle 4, Cycle 11) \quad (C_4C_{11} = X^3)
Cycle Pair 5: (Cycle 6, Cycle 11) \quad (C_6C_{11} = X^4)
Cycle Pair 6: (Cycle 7, Cycle 9) \quad (C_7C_9 = X^8)
Cycle Pair 7: (Cycle 7, Cycle 10) \quad (C_7C_{10} = X^7)
Cycle Pair 8: (Cycle 7, Cycle 11) \quad (C_7C_{11} = X^4)
Cycle Pair 9: (Cycle 8, Cycle 11) \quad (C_8C_{11} = X^2)
Cycle Pair 10: (Cycle 10, Cycle 11) \quad (C_{10}C_{11} = X^5)

Cycle Triple 1: (Cycle 4, Cycle 8, Cycle 11) \quad (C_4C_8C_{11} = X^4)
Cycle Triple 2: (Cycle 7, Cycle 10, Cycle 11) \quad (C_7C_{10}C_{11} = X^8)
Example (cont.)

\[ A(X) = \frac{X^{12} \cdot 1 + X^7(1 - X) + X^{11}(1 - X) + X^6(1 - 2X + X^2) + X^8 \cdot 1 + X^7(1 - X) + X^7(1 - X - X^4)}{1 - 2X - X^3} \]

\[ = \frac{X^6 + X^7 - X^8}{1 - 2X - X^3} \]

\[ = X^6 + 3X^7 + 5X^8 + 11X^9 + 25X^{10} + \cdots. \]
Other weight enumerators

a) Can describe the code parameters in more detail by enhancing the branch labels
   • Multiply label by $W^w$, where $w$ is the branch input weight
   • Multiply label by $L$; denoting that each branch has length 1

b) Input-output weight enumerating function (IOWEF)
   • $A(W,X,L) = \sum_{w,d,l} A_{w,d,l} W^w X^d L^l$
\[
\Delta = 1 - (X^8 W^4 L^7 + X^3 W^3 L^5 + X^7 W^3 L^6 + X^2 W^2 L^4 + X^4 W^4 L^7 + X^3 W^3 L^5 \\
+ X^3 W^3 L^3 + X W L^2 + X^5 W^2 L^4 + X^4 W^2 L^3 + X W L) + (X^4 W^6 L^7 \\
+ X^8 W^4 L^7 + X^3 W^3 L^6 + X^3 W^3 L^5 + X^4 W^4 L^7 + X^8 W^4 L^7 + X^7 W^3 L^6 \\
+ X^4 W^2 L^4 + X^2 W^2 L^3 + X^5 W^3 L^4) - (X^4 W^4 L^7 + X^8 W^4 L^7) \\
= 1 - X W (L + L^2) - X^2 W^2 (L^4 - L^3) - X^3 W^2 L^3 - X^4 W^2 (L^3 - L^4)
\]

\[
\sum_i F_i \Delta_i = X^{12} W^4 L^8 \cdot 1 + X^7 W^3 L^6 (1 - X W L^2) + X^{11} W^3 L^7 (1 - X W L) \\
+ X^6 W^2 L^5 [1 - X W (L + L^2) + X^2 W^2 L^2] + X^8 W^4 L^8 \cdot 1 \\
+ X^7 W^3 L^7 (1 - X W L) + X^7 W L^4 (1 - X W L - X^3 W^2 L^3) \\
= X^6 W^2 L^5 + X^7 W L^4 - X^8 W^2 L^5.
\]

\[
A(W, X, L) = \frac{X^6 W^2 L^5 + X^7 W L^4 - X^8 W^2 L^5}{1 - X W (L + L^2) - X^2 W^2 (L^4 - L^3) - X^3 W L^3 - X^4 W^2 (L^3 - L^4)} \\
= X^6 W^2 L^5 + X^7 (W L^4 + W^3 L^6 + W^2 L^7) \\
+ X^8 (W^2 L^6 + W^4 L^7 + W^4 L^8 + 2 W^4 L^9) + \ldots.
\]